Virtual Enterprise Formation with Agents – an Approach to Implementation

Sobah Abbas Petersen[1], Jinghai Rao[1], Mihhail Matskin[2]
[1]*Department of Computer and Information Sciences,*
Norwegian University of Science and Technology, 7491 Trondheim, Norway
Email: {sap, Jinghai}@idi.ntnu.no
[2]*Department of Microelectronics and Information Technology,*
Royal Institute of Technology, SE-164 40 Kista, Sweden
Email: misha@imit.kth.se

Abstract

The formation of a Virtual Enterprise and the selection of its partners is an important process in the lifecycle of a Virtual Enterprise. In this paper, we present the Virtual Enterprise formation process as an Agent Interaction Protocol and an approach to its implementation. We have focussed on the selection of partners within the formation process in order to understand these interactions and the contents of the messages that are exchanged between the agents. Based on this, we describe how the AGORA multi-agent architecture can be used to support the formation of a Virtual Enterprise.

1. Introduction

The formation of a Virtual Enterprise (VE) is an important phase in the lifecycle of a VE. The selection of the partners that will do the work in the VE is central to the formation phase and is one of the success factors for a VE, [2]. We consider a VE as *a team of partners that collaborate to achieve a specific goal*. The partners of a VE may be human beings, organisations and/or software agents.

In this paper, we present a multi-agent architecture, AGORA, to support the formation of VEs. We believe that software agents, (or agents), are a suitable means of representing the partners of a VE. One important reason is that by delegating the agents to conduct the negotiation on behalf of the partners, the partners could then have the time to do the actual work required in the current VE.

We have developed an agent-based Enterprise Model of a VE by analysing the entities in VEs, their relationships and how they can be used in an agent context. In our model, the VE has a goal, which is achieved by a set of activities, which are performed by a set of roles. The agents that fill these roles are the members of the VE and the agents are selected on the basis of how well they meet the requirements for the roles.

The idea of using agents to represent the partners in a VE is not new, e.g. [6], where an electronic market and auctions are used to select the partners of a VE. We have used AGORA to support the formation of VEs, within the context of an electronic market place, [5]. We describe the AGORA architecture as well as the architecture for a single agent to represent the partners in a VE. The interactions among the agents during the VE formation process is described as an Agent Interaction Protocol (AIP) and the VE formation scenario is illustrated using a simple hypothetical example.

The rest of this paper is organised as follows: Section 2 describes the VE formation process, Section 3 presents the model of a single agent, Section 4 describes the AGORA architecture, Section 5 illustrates how AGORA can be used to support VE formation using an example, Section 6 discusses the conclusions.

2. Virtual Enterprise Formation

VEs have a limited lifetime; thus they need to be formed very quickly in order to meet the deadlines of the goals and there is a need to form them often. An important part of the formation is the selection of partners, who are selected on the ability to fulfil the VE's requirements.

The agents in a VE can be classified as *VE Initiator* (who may also be the customer), who takes the initiative to form the VE, and *VE Partner* (who may also be the VE Initiator), who are the entities that form the VE. A VE Partner evolves from someone that is interested in becoming a part of the VE and submits a bid, *Interested Partner*, to someone that is considered for the VE and a contract is negotiated, *Potential Partner*, to someone who is actually a part of the VE.

The first step in the process of selecting partners for a VE is the alignment of the goals of the Interested Partners with the goals of the VE. The second step is matching the Interested Partners to the requirements of

the roles. The requirements are structured into skills and capabilities, availability and cost requirements. The third step is the verification of the information provided in the bids. This is to ensure that the Potential Partner indeed has the experience and the means of delivering to the VE as claimed. In reality, this is often conducted in the form of interviews and workshops. Once the verification is conducted, the VE Initiator and the Potential Partners (or the Potential Partners themselves) negotiate to agree upon the terms of the contract.

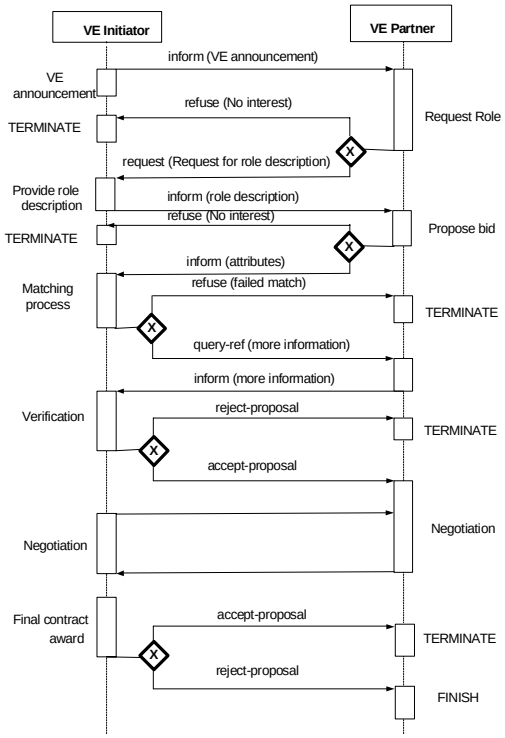

Figure 1: Agent Interaction Protocol for VE Formation

The selection process can be expressed as an AIP, the basic idea of which is similar to that of auctions and the Contract Net Protocol [3]. The AIP corresponding to the formation of a VE is shown in Figure 1.

3. Model of an Agent

The components of an agent's knowledge base required to support the selection process are its goals, activities and capabilities, which are described by a set of attributes. In addition, an agent has a rule base to support its decision making process and a plan to tell it what to do at any point in time. In our approach, we consider the same agent architecture for both the VE Initiator and the partners. This is because the VE Initiator does not always play the role of a "broker" only, but can also be a partner in the VE. The information that is represented by the goals, the activities and the capabilities of the VE Initiator and the VE Partners are slightly different and this is summarised in Table 1.

Table 1: Information represented by the Agent Model

Entity	VE Initiator	VE Partner
Goals	Goals of the VE	Goals of the partner
Activities	Activities that need to be performed to achieve the goals of the VE.	Set of experience of the partner.
Capabilities	Requirements (skills, time, costs, etc.) for the roles of the VE.	Work that the partner is capable of doing.

4. Forming Virtual Enterprises Using AGORA

In this section, we describe the AGORA multi-agent architecture and how it can be used to support VEs.

4.1. AGORA Multi-agent Architecture

AGORA is a multi-agent infrastructure which provides support for implementation of software agents and agent-based marketplaces, [5]. The central concept is that of an Agora node which is a cooperative node facilitating communication, coordination and negotiation among the agents. When an *Agora node* is created, the default agents are created and connected to the agora node automatically. They are the *Agora Manager* (for performing general management and matchmaking functions), *Coordinator* (for supporting a coherent behaviour between agents in the node) and *Negotiator* (for dealing with conflict resolution via negotiation).

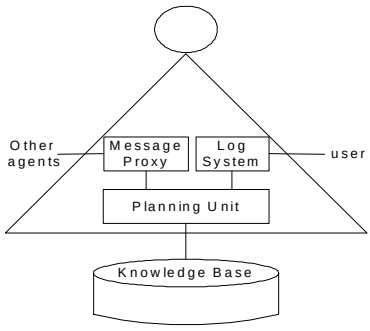

Figure 2: Structure of an Agent

In addition to Agora nodes and default agents, the system also has *registered agents.* In a general marketplace scenario, the registered agents can act as either buyers or sellers. The structure of a single agent, either a default agent or a registered agent is illustrated in Figure 2. An agent uses the *Message Proxy* and the

Log System to interact with the outside world, (e.g. the human user). It communicates with other agents using FIPA ACL, [4], and the FIPA messages are sent and received through the Message Proxy.

We use a Prolog-based presentation for messages, facts and rules in the *Knowledge Base*, implemented using the XProlog system, [1]. In order to integrate the FIPA messages with the Knowledge Base, a *Compiler* between FIPA messages and Prolog clauses is implemented.

The *Planning Unit* decides the agent's next action by a set of explicitly defined rules. In Agora, the plan is specified in a XML-based scripting language. Each step in the plan has an action to be performed and post-conditions. The action refers to an outgoing FIPA message or a method (function) written in Java or Prolog. Post-conditions are described as a reaction of the agent to a communicative act received from another agent.

4.2. Virtual Enterprises in AGORA

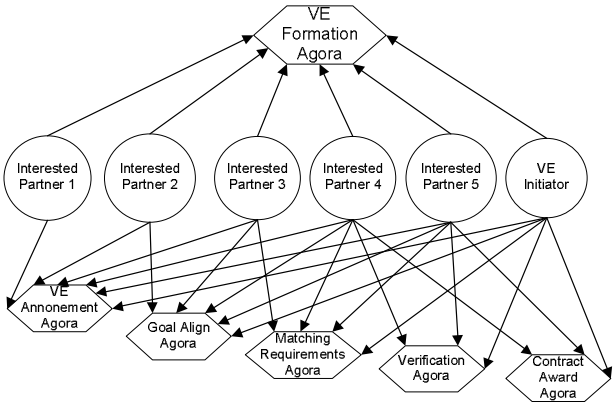

Figure 3: The AGORA Multi-agent Architecture

A description of how AGORA could be used to support the lifecycle of a VE was presented in [7]. The Agora nodes for VE Formation are presented in **Figure 3**. It contains a general Agora node (VE Formation) for the complete process and separate Agora nodes for each step in the VE formation process. Registered Agents (Interested Partners) can be registered in more than one Agora node. Each Agora node provides the right context for the specific step in the formation process, i.e. the right support and a meeting place for all the participants. Having a separate Agora node for each step ensures information security and privacy, e.g., an Interested Partner whose bid has been refused cannot register at the next Agora node.

5. Example

We use a simple example to illustrate the formation process and the selection of partners by matching agents to roles in a VE. Consider a VE formed to design and create an Intranet for a company. The main goal of the VE, "Create an Intranet", can be decomposed into two subgoals, "Design Intranet" and "Create Intranet". The two subgoals are achieved by performing the activities, "Design Intranet" and "Create Intranet". The two roles that are required for this VE are an "Intranet Designer" and a "Webpage Developer". The VE Initiator is looking for two partners that meet the requirements for these roles. In the rest of this section, we will describe how the formation process of this VE is implemented in AGORA.

5.1. VE Announcement

The VE announcement consists of the main goal of the VE and the set of roles that need to be filled (at VE Announcement Agora node).

```
ve_announcement(
     goal(create_intranet, 280203,40000),
     roles([intranet_designer,
            web_program_developer])).
```

If an agent is interested in performing any of the roles, it requests for more information on that specific role. The VE Initiator will then respond with the requirements for the requested role(s). Table 2 shows the set of requirements and the matching conditions for the role Webpage Developer.

Table 2: VE Requirements and Matching Conditions

Requirements	Range & Matching Conditions
Skills	HTML, JAVA, XML
Min. no. of skills required	>=2
Experience	>=2 years
Availability	Start_date<010103, end_date=<280203, 80% of the time, Matching condition: computed no. of hours =<300
Cost per hour	<60
Performance rating	Range: 1..10, >=6
Commitment	Range: 1..10, >7

The Interested Partners return bids after receiving the requirements for the roles. In the bids, the Interested Partners, (e.g. for the role of Webpage Developer), fill their values for required attributes as shown below:

```
bid_skill(programmer1,
  role(Webpage_developer),
  attributes(skills([java,xml,html]),
     experience_by_year(3),
     performance_rating(7),
     commitment(8)).
```

5.2. Matching Agent to Roles

The partner's goals are aligned with that of the VE if there is a goal in the VE's goal structure that matches that of the partner's (at the Goal Alignment Agora node). The matching is based on the attributes of the goal. The requirements for the roles are structured into skills, availability and cost requirements. The matching process consists of matching first the skills, then the availability and finally the costs (at the Matching Requirements Agora node). If the Interested Partner meets all the requirements for the role, s/he becomes a Potential Partner and the VE Initiator would now like to verify if s/he actually does have the experience claimed in the bid. Thus, the VE Initiator requests for the Potential Partner's activities or experience structure (at the Verification Agora node). The VE Initiator informs the Potential Partners whose bids are rejected and a contract is signed with the Potential Partners whose bids are accepted (at the Contract Award Agora node).

5.3. Agent Plans

An agent's plan consists of all the communication exchange and protocols. The part of the VE Initiator's plan file that corresponds to the process of matching an agent's skills to the requirements is shown below:

```
<step>
  <id>match_skill</id>
  <action>match_skill</action>
  <case>
    <postcondition>
      <performative>inform</performative>
      <ontology>VE</ontology>
    </postcondition>
  <nextstep>match_availability</nextstep>
  </case>
  <case>
    <postcondition>
      <performative>refuse</performative>
      <ontology>VE</ontology>
    </postcondition>
  <nextstep>TERMINATE</nextstep>
  </case>
</step>
```

Some steps in the plan file refer to agent actions which are presented separately in action files. There are two kinds of bodies in an action: *wrapper* for a message that can be predefined, and *implementedBy* for a message that cannot be predefined. The part of the action file for the VE Initiator for matching the skills is shown below:

```
<action>
  <id>match_skill</id>
  <implementedBy>
    <code>
      <codeLanguage>xprolog</codeLanguage>
      <codeMethod>match_skill</codeMethod>
    </code>
  </implementedBy>
</action>
```

6. Conclusions

This paper describes how the AGORA multi-agent architecture can be used to support the formation of VEs. The partners in a VE are represented by agents and the VE formation process is expressed as an AIP. We have used a simple example to illustrate our approach and to describe the implementation. We have not addressed the negotiation process(es) during the VE formation although we believe that this is an important part of the process. We plan to extend our work to include a more detailed description of the VE requirements and to support the automatic translation of AIPs to agents' plans.

7. Acknowledgements

This work is partially supported by the Norwegian Research Foundation and the ADIS project.

8. References

[1]. http://www.iro.umontreal.ca/~vaucher/xprolog/

[2]. Camarinha-Matos, L. M., H. Afsarmanesh, "Virtual Enterprise Modelling and Support Infrastructures: Applying Multi-agent System Approaches", in *Multi-agent Systems and Applications*, M. Luck, V. Marik, O. Stpankova, R. Trappl (Eds.), LNAI 2086, Springer, 2001.

[3]. Davis, R., R. G., Smith, "Negotiation as a Metaphor for Distributed Problem Solving", A. H. Bond and L. Gasser (Eds.) *Readings in Distributed Artificial Intelligence*, Morgan Kaufmann Publishers, 1988, pp. 333-356.

[4]. FIPA ACL http://www.fipa.org/

[5]. Matskin, M., O. J. Kirkeluten, S. B. Krossnes and Ø. Sæle. "Agora: An Infrastructure for Cooperative Work Support in Multi-Agent Systems", T. Wagner, O. Rana (Eds.) *Infrastructure for Agents, Multi-Agents, and Scalable Multi-Agent Systems*. Springer Verlag, LNCS, Volume 1887, 2001, pp.28-40

[6]. Oliveira, E. and A. P. Rocha, "Agents' Advanced Features for Negotiation in Electronic Commerce and Virtual Organisation Formation Process", *European Perspectives on Agent Mediated Electronic Commerce*, Springer Verlag, June 2000.

[7]. Petersen, S. A., M. Divitini, M. Matskin, "An Agent-based Approach to Modelling Virtual Enterprises", *Int. Journal of Production Planning & Control*, Vol. 12, No. 3, Issue Apr. 2001, pp. 224-233.